Anotación automática de personas en programas de TV sin supervisión

Ingeniería de Sistemas Audiovisuales

Trabajo Final de Grado
Presentado por
Joshua Molina Torrell

Escuela Superior de Ingeniería Industrial, Aeroespacial y Audiovisual de Terrassa
Universidad Politécnica de Catalunya

Dirigido y Supervisado por: Josep Ramon Morros Rubió

10 DE MAYO DE 2018
Terrassa

Descripción

La gran cantidad de datos visuales que se generan en la actualidad lleva a la necesidad de crear herramientas de anotación que permitan la búsqueda y la recuperación de la información que queramos en los vídeos. Una de las informaciones más importante de un vídeo es la identidad de personas. En este contexto, la anotación consiste en determinar quién aparece y cuando lo hace.

Agradecimientos

Con motivo del Trabajo Final de Grado, me gustaría agradecer en primer lugar a la Universidad Politécnica de Catalunya, especialmente a la ESEIAAT, por el trato recibido durante todos estos años que he estado realizado el grado en Ingeniería de Sistemas Audiovisuales.

Por otra parte, me gustaría agradecer todo el esfuerzo que ha dedicado el tutor de este proyecto, Josep Ramon Morros. Me ha estado guiando durante el desarrollo de este proyecto, además de los consejos recibidos. Sinceramente, muchas de las ocasiones en las cuales he tenido un problema, si él no hubiese estado ahí no habría sido posible.

También agradecer al equipo que da soporte al servidor de imagen y video, y especialmente a Albert Gil, el cual cuando he tenido algún problema de acceso no ha tenido ningún impedimento en ponerse en contacto conmigo.

Historial de revisiones

Nº de Revisión	Fecha	Descripción
0	20/04/2018	Creación del documento
1	07/05/2018	Revisión del documento
2	09/05/2018	Revisión del documento

Tabla 1 Historial de revisiones

Distribución del documento

Nombre	Correo electrónico
Joshua Molina Torrell	joshuamolina1994@gmail.com
Josep Ramon Morros Rubió	ramon.morros@upc.edu

Tabla 2 Distribución del documento

Escrito por:		Revisado y aprobado por:	
Fecha	20/04/2018	Fecha	09/05/2018
Nombre	Joshua Molina Torrell	Nombre	Josep Ramon Morros Rubió
Rol	Autor del proyecto	Rol	Tutor del proyecto

Tabla 3 Autor y Tutor del documento

Tabla de Contenidos

Introducción

1

1. Introducción

1.1 Contextualización del proyecto

La gran cantidad de datos visuales que se generan en la actualidad lleva a la necesidad de crear herramientas de anotación que permitan la búsqueda y la recuperación de la información que queramos en los vídeos. Una de las informaciones más importante de un vídeo es la identidad de personas. En este contexto, la anotación consiste en determinar quién aparece y cuando lo hace.

La Universidad Politécnica de Catalunya (UPC) participa cada año en un proyecto europeo llamado *Camomile*[14]. Este proyecto tiene como finalidad mejorar las anotaciones automáticas que se basa en datos *3M (Multimodal, Multimedia y Multilenguaje)*, ya que puede ser útil para la ciencia de la informática pero también para la ciencia social. En este proyecto participan diferentes organizaciones internacionales, y las evaluaciones se llevan a cabo a través de la iniciativa *MediaEval*[8] *(Multimodal Person Discovery in Broadcast TV Task)*.

MediaEval es una iniciativa en la cual se evalúa nuevos algoritmos para obtener información multimedia, como por ejemplo reconocimiento de voz, análisis de contenido multimedia, música y audio, redes sociales, etc. Principalmente se basa en aspectos humanos y sociales. Los participantes deben proporcionar para cada vídeo, una lista de personas que hablan y aparecen al mismo tiempo. La forma de identificar a estas personas es a través del audio, por ejemplo, mediante la transcripción de voz, o visual, por ejemplo, utilizando un reconocimiento óptico de caracteres.

La iniciativa de *MediaEval* proviene de otra iniciativa llamada REPÈRE[16], que se basaba en el reconocimiento *multimodal* de personas en emisiones de televisión.

La diferencia entre esta iniciativa y *MediaEval* es que en esta última solo se admiten el uso de algoritmos no-supervisados, esto quiere decir, algoritmos que no tengan etiquetas ni modelos ya existentes.

En el primer año que comenzó esta iniciativa, en el año 2015, la Universidad Politécnica de Catalunya (UPC) participó por primera vez.

El desarrollo que se va a realizar en este trabajo se basa en intentar mejorar la parte de extracción de nombres. Esta parte está dentro de un bloque (OCR + NER) dentro del diagrama de bloques desarrollado por la UPC.

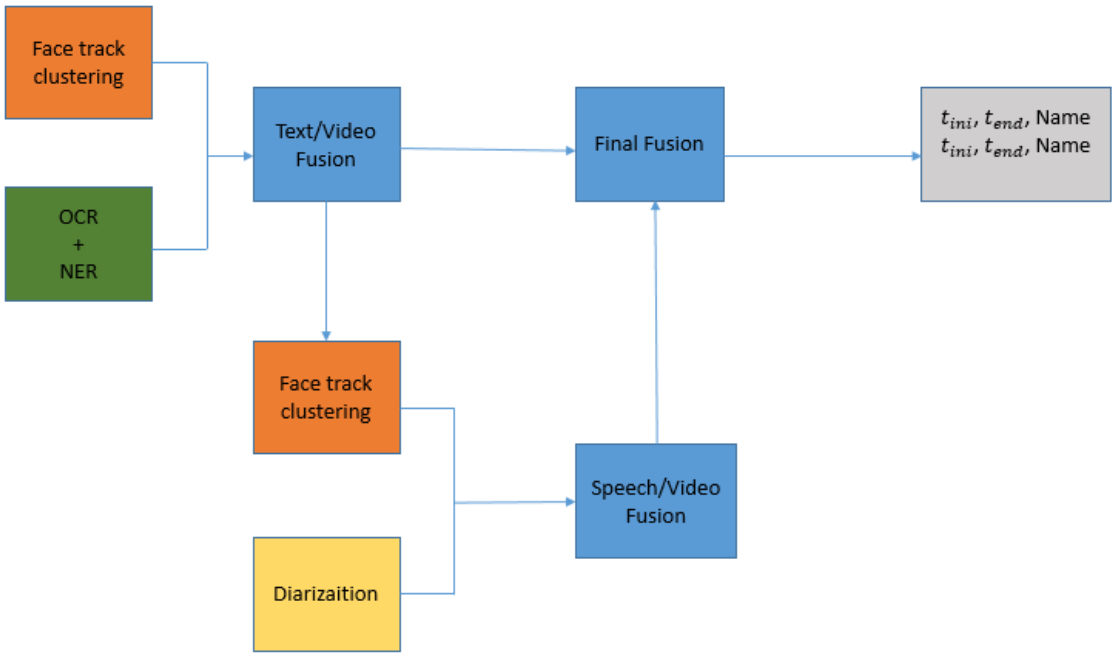

Ilustración 1 Diagrama de bloques desarrollado por la UPC

1.2 Objetivos

El objetivo de este proyecto se basa en mejorar la implementación de un sistema de extracción de nombres, al desarrollo ya existente. Otro de los objetivos es realizar pruebas de los algoritmos ya existentes en el corpus de *MediaEval 2015*[9] y *MediaEval 2016*[10].

1.3 Requerimientos y Especificaciones

Los requerimientos de este proyecto, como se ha especificado anteriormente son:

- Diseñar e implementar nuevos algoritmos para la extracción de nombres de personas en emisiones de televisión.

Por otro lado, las especificaciones de este proyecto son las siguientes:

- Para el desarrollo de los algoritmos se utilizará el lenguaje de programación *Python*.
- Evaluar los algoritmos implementados en el corpus de *MediaEval 2015* y *MediaEval 2016*.
- Para poder evaluar los resultados, se utilizará la métrica de evaluación *Mean Average Precision*[28] (MAP)

1.4 Plan de trabajo

1.4.1 Work Packages

WP1. Propuesta del proyecto y redacción del plan de trabajo

WP2. Lectura del estado del arte

WP3. Investigación de diferentes algoritmos para la extracción de nombres (NER)

WP4. Desarrollo del algoritmo *Stanford NER*[18]

WP5. Desarrollo de un algoritmo a través de diferentes filtros para la extracción de nombres (NER)

WP6. Evaluación de los resultados y propuesta de mejoras

WP7. Redacción de la memoria final

1.4.2 Justificación

Según los resultados obtenidos hace dos años, se prioriza los bloques que tenían unos resultados más flojos. Por eso se ha decidido la implementación de nuevos algoritmos que ayuden a mejorar los resultados dentro del bloque del *Named Entity Recognition*. Durante el proyecto se intentó mejorar la detección de texto, e implementar un algoritmo diferente al ya implementado, pero por problemas externos no se pudo implementar.

1.4.3 Desviaciones respecto al plan de trabajo inicial

El plan de trabajo que se estableció inicialmente repartía el tiempo necesario para elaborar los bloques que se han especificado en los puntos anteriores. Cuando se comenzó a realizar las pruebas del desarrollo que ya estaba implementado de años anteriores, se encontró que había que realizar algunos cambios, ya que las estructuras tanto de carpetas como de nomenclaturas en las bases de datos habían cambiado. Una vez funcionando, se encontró varios algoritmos para comenzar a desarrollar la implementación en el bloque que hace referencia a la extracción de nombres.

A medida que iba avanzando el proyecto, se propuso realizar una implementación para mejorar el bloque de detección de texto y que sustituyese a la técnica *LOOV*[29]. Se intentó utilizar unas funciones del siguiente repositorio de *Github: Deep Video Analitycs*[30]. A medida que estábamos probando de implementar este desarrollo, el administrador del repositorio hizo varios cambios en sus funciones e intentamos contactar con él para que nos diese una solución, pero no fue proporcionada. Este tiempo que perdimos hizo que nos retrasase en el plan de trabajo.

1.4.4 Diagrama de Gantt

Este es el diagrama de Gantt que está asociado al proyecto:

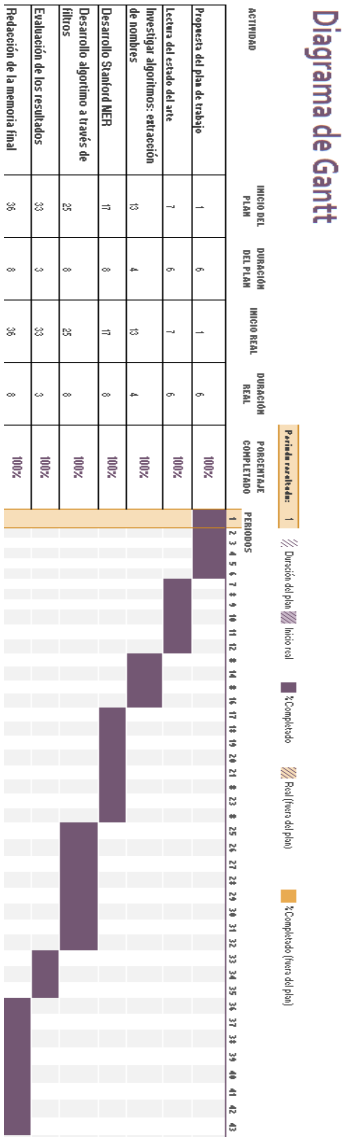

Ilustración 2 Diagrama de Gantt

Estado del Arte

2

2. Estado del Arte

Como se ha mencionado en puntos anteriores, desde que se inició con las iniciativas, en el año 2011, con *REPÈRE* hasta hoy con *MediaEval,* con el fin de mejorar en la investigación que se basa en la tecnología de anotación automática para vídeo se ha avanzado mucho.

Las preguntas principales que se hacen en estas iniciativas son: ¿Quién aparece en los vídeos? ¿Quién está hablando?

En nuestro proyecto, nos basaremos principalmente en la primera pregunta. Utilizaremos técnicas de aprendizaje no-supervisado para el desarrollo de nuevos algoritmos que ayuden a mejorar el prototipo ya existente.

Este aprendizaje no-supervisado es un algoritmo que se basa en poder dar resultados a partir de un conjunto de datos que están sin etiquetar, es decir a medida que vamos leyendo datos vamos almacenando la información que queremos.

2.1 Detección del texto

Este proyecto utiliza una técnica de detección/reconocimiento de texto que ya se implementó en el anterior proyecto. Esta técnica es la llamada *LOOV,* la cual utiliza el método llamado *Optical Character Recognition*[2] (OCR).

2.1.1 LOOV

Esta técnica presenta un sistema OCR que detecta y reconoce los textos que están solapados en el vídeo. Finalmente hace un post-procesamiento para combinar múltiples transcripciones de la misma caja de texto. Esta técnica se basa exclusivamente en la información que da la imagen.

A continuación se explica el procedimiento que utiliza para la detección de texto:

- Se aplica un filtro Sobel vertical y horizontal para poder detectar los contornos de los caracteres. Se aplica unas cuántas iteraciones de dilataciones para poder reconocer los caracteres que están consecutivos.
- Una vez realizado el primer paso, se aplica una obertura para poder separar cada palabra. Para poder detectar cada una de las líneas de texto se utiliza un método llamado *proyecciones horizontales y verticales,* lo que conlleva a una gran cantidad de falsos positivos.
- Para poder filtrar estas falsas alarmas, se aplica una detección local.
- Para el seguimiento temporal del texto se utiliza un filtro temporal, esta información temporal que da el filtro se utiliza para filtrar los falsos positivos y además, poder recuperar las cajas dónde la detección local falla.

Una vez se ha detectado la localización del texto en la imagen, se realiza el paso de la extracción de textos. Pero antes de realizar el paso de la extracción, se mejora la resolución de la imagen a través de una interpolación bi-cúbica, y se aplica una binarización a la imagen con un umbral que se calcula a partir del algoritmo de *Sauvola*.

2.1.2 CTPN

La detección de texto basada en CTPN[26] (Connection Text Proposal Network) localiza con precisión líneas de texto en imágenes y vídeos. Uno de los problemas es detectar la longitud de la línea de texto, por lo que a continuación se expone la solución que da esta técnica:

La idea principal es predecir la posición vertical del texto, por lo que utiliza un marco vertical que predice la ubicación de dónde se encuentra el texto, este marco vertical tiene una anchura fija y eso ayuda a mejorar la precisión de la ubicación de toda la línea de texto.

Una vez se ha encontrado la ubicación de la línea de texto hay un proceso de ventanas deslizantes, y un algoritmo de construcción de las líneas de texto basado en gráficos para fusionar los segmentos de texto obtenidos con el marco vertical.

Se basa en propuestas secuenciales que están conectadas por una red neuronal recurrente.

Este método es el que utiliza el repositorio de github *Deep Video Analytics.*

2.2 Extracción de nombres

Para la extracción de nombres se utiliza una herramienta llamada *Named Entity Recognition (NER),* que trata de localizar y clasificar en categorías, como personas, organizaciones, lugares, etc, las entidades encontradas en un texto.

Para el desarrollo del proyecto se han contemplado la implementación de estas dos herramientas:

- Freeling[5]
- Stanford NER[18]

2.2.1 Freeling

FreeLing es una biblioteca C++ que permite el análisis del lenguaje, y está diseñado para utilizarse como una biblioteca externa. Si se llama desde una aplicación/programa que esté hecho en Java o Python se debe hacer a través de una API.

Freeling codifica la información morfológica en etiquetas *PoS (Part-os-Speech*) que se basa en las propuestas que hace *EAGLES*.

La propuesta que lleva a cabo EAGLES es codificar todas las características morfológicas de la mayoría de idiomas europeos. Lo hace de manera que todas las etiquetas *EAGLES PoS* son de longitud variable. Cada carácter corresponde a una característica morfológica, como en el siguiente ejemplo, para la categoría *nombre*:

Position	Attribute	Values
0	category	**N**:*noun*
1	type	**C**:*common*; **P**:*proper*
2	case	**N**:*nominative*; **G**:*genitive*; **D**:*dative*; **F**:*accusative*;
3	gen	**F**:*f*; **M**:*m*; **C**:*c*
4	num	**S**:*s*; **P**:*p*; **N**:*n*

Ilustración 3 Análisis morfológico en el algoritmo Freeling

Todo este análisis morfológico es posible gracias a alguna de las siguientes funciones:

- Identificación del lenguaje
- *Tokenization*
- División de frases
- Análisis morfológico
    - Detección de números
    - Detección de fechas
    - NER
- *PoS* (*Part-of-Speech*)

Los idiomas que soporta esta herramienta son los siguientes:

- Dialecto Asturiano
- Catalán
- Galés
- Alemán
- Inglés
- Español
- Francés
- Gallego
- Croata
- Italiano
- Noruego
- Portugués
- Ruso
- Esloveno

Para nuestro proyecto nos centraremos en los siguientes idiomas: inglés, catalán, español, francés y alemán. Los diccionarios de estos idiomas consisten en:

- *Catalán:* Contiene más de 520000 formas correspondientes a 71,000 lemas
- *Alemán:* Contiene cerca de 395000 formas correspondientes a 130000 lemas
- *Inglés:* Contiene 68000 formas correspondientes a 37000 lemas
- *Español:* Contiene 555000 formas correspondientes a 76000 lemas
- *Francés:* Contiene 54000 formas correspondientes a 54000 lemas

2.2.2 Stanford NER

Stanford NER es un reconocedor de nombres implementado en el lenguaje JAVA. Este reconocedor utiliza el método por etiquetas, de las cuales lleva a cabo estas tres clases:

- Persona
- Organización
- Lugar

El algoritmo que utiliza es el llamado muestreo de Gibbs, lo utilizan para añadir dependencias que no son locales para secuenciar modelos para la extracción de información.

Este sería un ejemplo del método:

Ilustración 4 Análisis morfológico en el algoritmo de Stanford NER

Los idiomas que reconoce esta herramienta son:

- Árabe
- Chino
- Inglés
- Francés
- Alemán
- Español

2.3 Evaluación de Resultados

La métrica que se va a utilizar para comprobar los resultados es la llamada *Mean Average Precision (MAP)*. Cuando aplicamos este método en la anotación de personas en la secuencia de un vídeo, el objetivo es determinar cuántas veces aparece una persona en una determinada secuencia.

A continuación se explica en que consiste este método:

Si hay m pistas que representan a una cierta persona y queremos predecir cuántas n apariencias pertenecen a la identidad I, entonces el *Average Precision* para este ejemplo sería el siguiente:

$$ap@n = \sum_{k=1}^{n} \frac{P(k)}{min(m,n)}$$

Dónde *P(k)* hace referencia a las pistas que realmente pertenecen a la identidad *I* hasta la posición *k*, *m* es la cantidad de nodos relevantes y *n* es el número de nodos predichos.

Para poder entender mejor esta técnica se muestra unos cuántos ejemplos:
Se predice un máximo de 5 apariciones de una persona en una secuencia de vídeo, pero en realidad solo aparece en las siguientes pistas *T1*, *T2*, *T4* y *T8*.

Ejemplo 1

T1, T3, T4, T7, T8

k=1 → P(1) = 1 / 1 = 1

k=2 → P(2) = 0 (Track 3 no representa a una persona)

k=3 → P(3) = 2 / 3

k=4 → P(4) = 0 (Track 7 no representa a una persona)

k=5 -->P(5) = 3 / 5

MAP@5 = (1 + 2 / 3 + 3 / 5) / 4 = 0.57

Ejemplo 2

T1, T4, T8, T3, T7

k=1 → P(1) = 1 / 1 = 1

k=2 → P(2) = 2 / 2 = 1

k=3 → P(3) = 3 / 3 = 1

k=4 → P(4) = 0 (Track 3 no representa a una persona)

k=5 -->P(5) = 0 (Track 7 no representa a una persona)

MAP@5 = (1 + 1 + 1) / 4 = 0.75

El orden se ha de tener en cuenta, ya que como se puede comprobar en los dos ejemplos da resultados distintos. Para que la técnica *MAP* sea mejor, los resultados más relevantes deben estar en las primeras posiciones como en el segundo ejemplo. Este orden solo afecta si al menos hay una predicción incorrecta, ya que si todas son correctas no importa el orden en el que estén. Al igual que si las predicciones incorrectas están al final, tampoco tiene importancia.

Desarrollo

3

3. Desarrollo del Proyecto

3.1 Base de datos

Los datos de entrenamiento que se utilizarán corresponden a la base de datos que se utilizó en la edición del año 2016, llamada *6 Months of Broadcast News de INA*.

Para el desarrollo del proyecto se utilizará es el subconjunto de videos siguientes:

- Un subconjunto de vídeos del corpus *INA* formado por una semana completa de grabación de tres canales de la televisión francesa.
- Un subconjunto de vídeos del corpus *DW/EUMSSI*
- Un subconjunto de vídeos del corpus de la televisión catalana *3-24*

Ilustración 5 Imagen de un vídeo del corpus DW/EUMSS

I

Ilustración 6 Imagen de un vídeo del corpus de la televisión 3-24

Ilustración 7 Imagen de un vídeo del corpus INA

3.2 Extracción de nombres

Para poder analizar/detectar si corresponde a un nombre de persona o no, la información extraída por el proceso que lleva a cabo la herramienta *LOOV*, se realiza diferentes pruebas con las técnicas *Freeling* y *Stanford NER*.

Freeling

Este detector de nombres se comporta de una manera muy parecida al *Stanford NER*, excepto que *Freeling* puede analizar texto en el lenguaje catalán.

A continuación se muestra unas ilustraciones de las diferentes pruebas que se han realizado para comprobar el comportamiento de la herramienta:

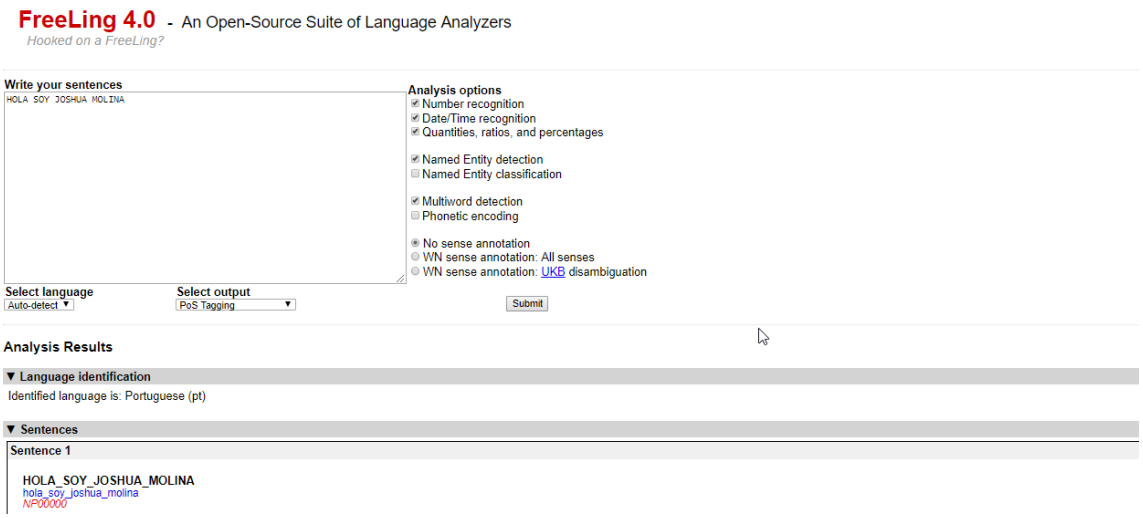

Ilustración 8 Ejemplo 1 Freeling

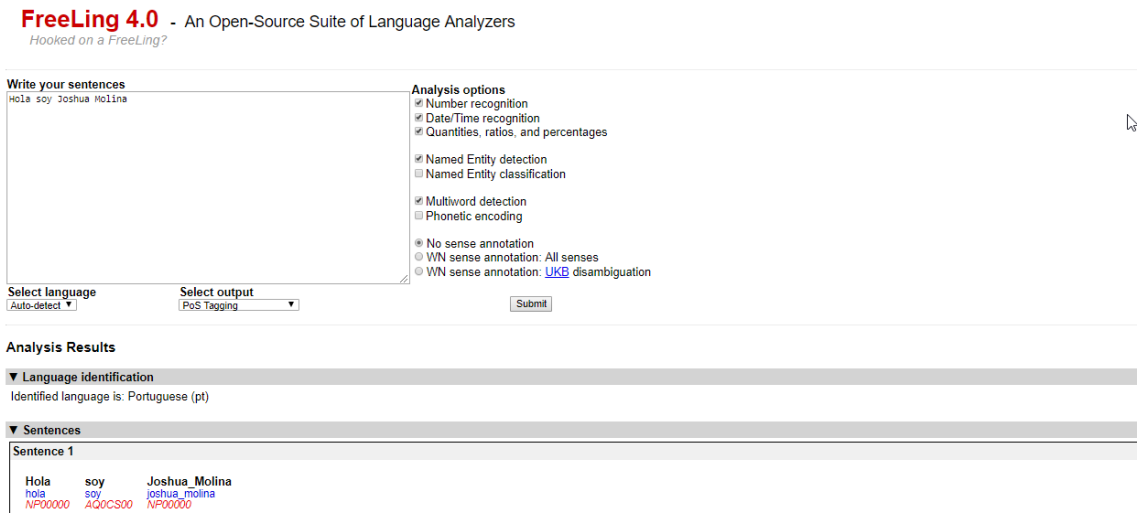

Ilustración 9 Ejemplo 2 Freeling

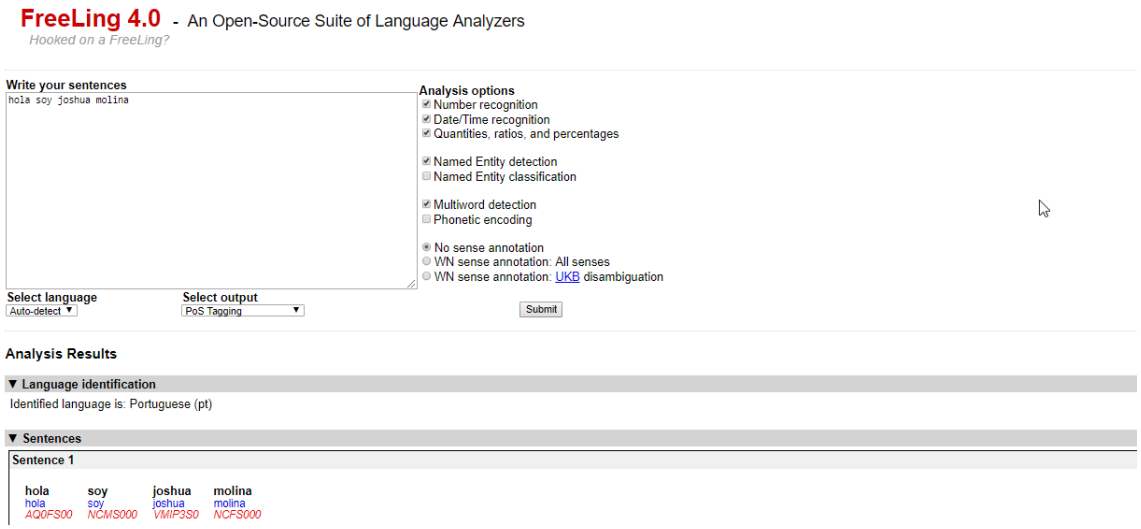

Ilustración 10 Ejemplo 3 Freeling

En el primer caso, el texto está escrito todo en mayúsculas y podemos observar como el resultado que indica la herramienta es que interpreta toda la frase como un *nombre propio*. *[Ilustración 8]*

En el segundo caso, el texto está escrito como estaría en un contexto normal, es decir de manera que las mayúsculas y las minúsculas se utilizan de manera correcta. En este caso, el resultado de la herramienta indica como *nombre propio* todo lo que comienza por mayúscula. *[Ilustración 9]*

En el tercer caso, el texto está escrito todo en minúscula, en este caso el resultado que indica *Freeling* es correcta, excepto cuando analiza el nombre, ya que no lo detecta como tal. *[Ilustración 10]*

Stanford NER

Con esta herramienta se realizan las mismas pruebas que con la herramienta *Freeling*.

A continuación se muestra unas ilustraciones de las diferentes pruebas que se han realizado para comprobar el comportamiento de la herramienta:

Ilustración 11 Ejemplo 1 Stanford NER

Ilustración 12 Ejemplo 2 Stanford NER

Ilustración 13 Ejemplo 3 Stanford NER

En el primer caso, el texto está escrito todo en mayúsculas y podemos observar como el resultado que indica la herramienta es correcta, y por lo tanto detecta bien el nombre propio. *[Ilustración 11]*

En el segundo caso, el texto está escrito como estaría en un contexto normal, es decir de manera que las mayúsculas y las minúsculas se utilizan de manera correcta. En este caso, el resultado de la herramienta es correcta, y por lo tanto detecta bien el nombre propio. *[Ilustración 12]*

En el tercer caso, el texto está escrito todo en minúscula, en este caso el resultado que indica es correcta, y por lo tanto detecta bien el nombre propio. *[Ilustración 13]*

Una vez analizado los resultados de las pruebas que se han realizado con las diferentes herramientas, se puede llegar a la conclusión que la herramienta *Freeling* no es válida para el desarrollo del proyecto. Se utilizará la herramienta *Stanford NER*.

3.2.1 Algoritmo Stanford NER

La herramienta *Stanford NER,* como ya se ha comentado anteriormente analiza un texto y etiqueta cada palabra en cuatro categorías:

- Persona
- Lugar
- Organización
- Desconocido

En el desarrollo del proyecto se ha implementado este algoritmo dentro de un script llamado *check_names_stanford.py,* en la función llamada *read_text_annotations_file,* que hace referencia a la salida del fichero que ha generado la técnica del *LOOV*.

Se llama a través de un comando JAVA, ya que es una librería externa. Este comando analizará un fichero de entrada, el cual ya habrá pasado por el proceso de *normalización del texto* y la salida será otro fichero en el cual cada palabra estará etiquetada según su categoría.

Después se le aplicará un filtro para que busque solo las etiquetas que hagan referencias a personas y así obtener otro fichero en el cual haya únicamente nombres de personas.

En el siguiente diagrama se muestra un ejemplo de cómo se realiza este proceso:

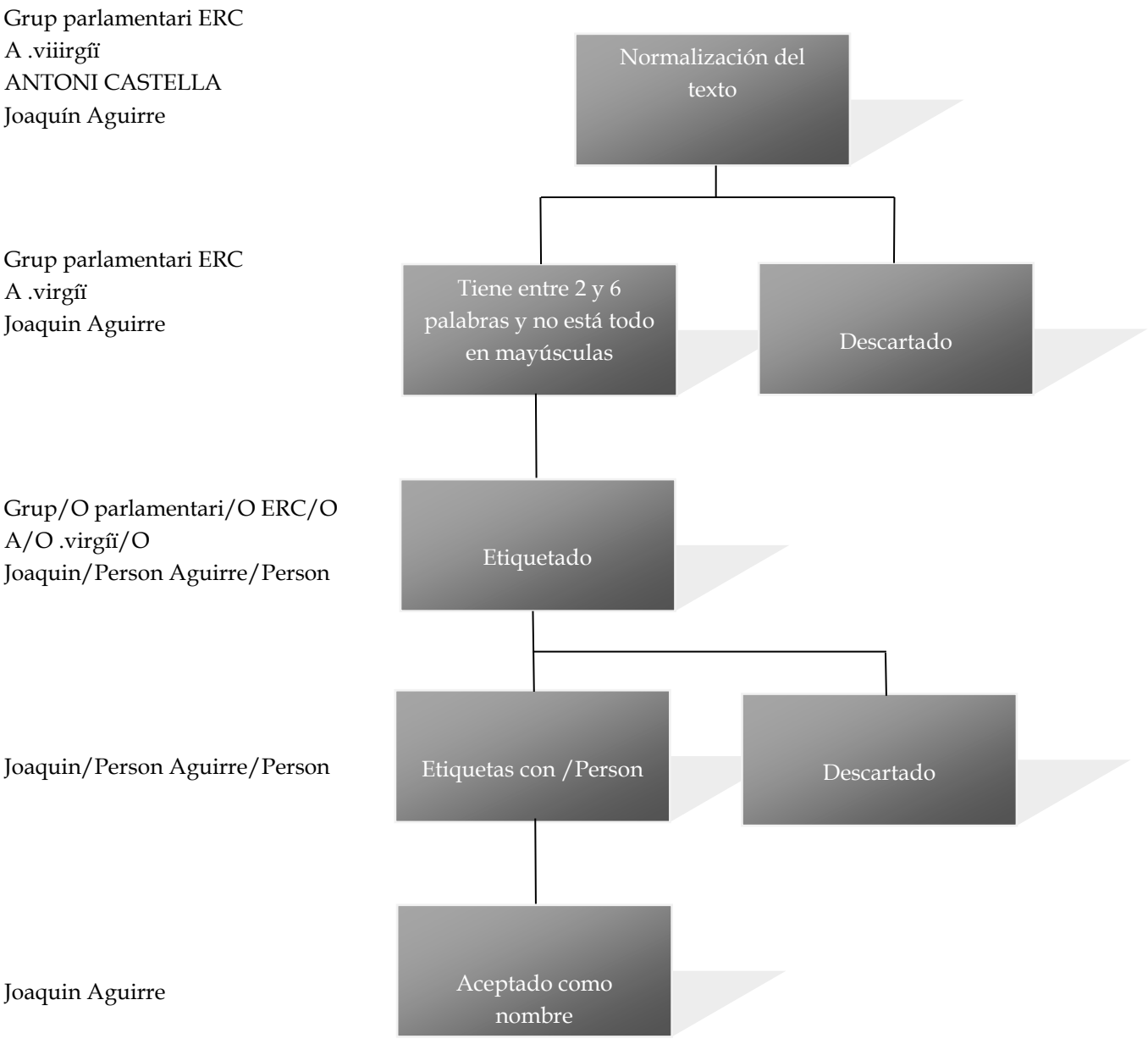

Ilustración 14 Proceso extracción de nombres con Stanford NER

3.2.2 Algoritmo basado en diccionarios

Este algoritmo que se ha desarrollado se basa en la mejora de diferentes bloques que ya existían.

3.2.2.1 Normalización del texto

En el bloque llamado *text_normalization* ya estaba implementado la eliminación o sustitución de todos los caracteres que podían llevar al error, como por ejemplo los acentos o en el caso del catalán la letra *ç*. Por lo que implementaron la:

- Eliminación de todos los acentos
- Convertir los caracteres no-ASCII a su equivalente en ASCII

Se ha implementado una nueva normalización, ya que revisando los resultados del año anterior aparecía bastantes veces nombres con caracteres repetidos consecutivos. Por ejemplo:

Joshua Mooooolina.

Para normalizar este error se ha implementado una función en la cual si encuentra un carácter repetido más de dos veces, filtra todos los caracteres repetidos dejando solo uno. Como en el ejemplo anterior quedaría de la siguiente forma:

Input	Output
Joshua Mooooolina	Joshua Molina

Tabla 4 Normalización de caracteres repetidos consecutivos

3.2.2.2 Revisión/Mejora de diccionarios de nombres

Los diccionarios que se crearon manualmente en el proyecto anterior y corresponden a estos nombres de ficheros:

- *First_names.txt* (Nombres de personas)
- *Last_names.txt* (Apellidos de personas)
- *Neg_list.txt* (Estaciones del año, nombres de meses y días de la semana…)
- *Neutral_list.txt* (Preposiciones, artículos…)
- *Pos_list.txt* (Nombres de oficio)

Se han revisado uno por uno. De los dos primeros ficheros de la lista han sido modificados muchos nombres y apellidos que eran incorrectos y se han actualizado de la forma correcta. En los ficheros restantes se han añadido más palabras con tal de poder hacer más efectiva la búsqueda de nombres de personas.

4 Resultados

4. Resultados

Todos los resultados que se han obtenidos han sido obtenidos a partir de los ficheros de referencia del proyecto *Multimodal Person Discovery in BroadCast TV de MediaEval 2016*

Cuando se comenzó el desarrollo del proyecto, la primera prueba que se realizó fue cambiar uno de los parámetros llamados *threshold* en el script que lanzaba la extracción de nombres llamado *extract_names.sh*. El valor por defecto que tiene este parámetro es de 0.75, esto quiere decir que de las palabras que reconoce como candidato a nombre coinciden 3 de cada 4 (75%) con los diferentes diccionarios de nombres que tenemos. Se probó este parámetro con los siguientes valores: 0.6, 0.8, 0.9 y 1.

- Con el valor 0.6 los resultados extraídos presentaban muchos falsos positivos, es decir, el filtro extraía demasiados resultados como nombres cuando en realidad no lo eran
- Con el valor 0.8 los resultados eran igual que con el valor 0.75
- Con el valor 0.9 y 1 los resultados extraídos presentaban muchos falsos negativos, es decir, el filtro no extraía la mayoría de nombres y en la mayoría de caso la salida de los ficheros estaba vacía

Una vez finalizado el desarrollo de los dos algoritmos que se han mencionado en puntos anteriores, la métrica que se ha utilizado para evaluar los resultados ha sido el *Mean Average Precision* (MAP), que se ha explicado en el *Estado del Arte,* tanto la del año 2015 como la del año 2016.

Se ha calculado el *MAP* para los resultados de los corpus *DW* y *3-24* que han sido obtenidos a partir de los dos algoritmos desarrollados: *Extracción de nombres a partir de un nuevo filtro y basado en los diccionarios revisados*, y el *Extracción de nombres a partir del algoritmo de Stanford NER*.

Los resultados se muestran en la siguiente tabla:

	MAP
Extracción de nombres a partir de un nuevo filtro y basado en los diccionarios revisados	9.65%
Stanford NER	9.87%

Tabla 5 Resultados MAP

Dado estos resultados tan bajos que no los esperábamos, se ha realizado un análisis más profundo que se explica a continuación:

Primer análisis

Se compara manualmente los ficheros *OCR* que pertenecen a la salida de la detección de texto de los distintos vídeos de las diferentes bases de datos, que se obtiene a través de la herramienta LOOV con los ficheros que contienen los resultados finales después de aplicarle los algoritmos que han sido desarrollados para comprobar si realmente lo que están funcionado bien o no. Se han analizado los 18 vídeos del corpus *3-24*, 30 vídeos al azar del corpus *DW*, y 20 vídeos del corpus *INA*.

A continuación se detallan los resultados obtenidos después de realizar este análisis:

Para el primer algoritmo desarrollado, *Extracción de nombres a partir de un nuevo filtro y basado en los diccionarios revisados*, se ha analizado que para los vídeos del corpus *3-24* y *DW* funciona bastante bien, ya que en los resultados de estos vídeos se puede observar que la mayoría de resultados pertenecen a nombres de personas y hay pocos falsos positivos y pocos falsos negativos. Sin embargo para el corpus de *INA* se obtienen muchos falsos positivos.

A continuación se muestran una media estadística realizada manualmente de todos los corpus:

Nombres extraídos	Resultado (%)
1095 filtrados totales	100%
278 falsos positivos (aprox)	25.38%
115 falsos negativos (aprox)	10.5%
817 filtrados correctos excepto falsos negativos	64.11%

Tabla 6 Resultados manuales del algoritmo basado en un nuevo filtro y diccionarios

Para el segundo algoritmo, *Stanford NER*, se ha realizado el mismo análisis que en el caso anterior y se ha obtenido que hace una mejor extracción de nombres con respecto el primer algoritmo para los tres corpus, ya que no contiene tantos falsos positivos.

Nombres extraídos	Resultado (%)
1513 filtrados totales	100%
200 falsos positivos (aprox)	13.21%
115 falsos negativos (aprox)	7.6%
1313 filtrados correctos excepto falsos negativos	79.1%

Tabla 7 Resultados manuales del algoritmo Stanford NER

Segundo análisis

Se compara manualmente el fichero de referencia del año 2016 que se utiliza para realizar el *MAP* con los ficheros OCR que se han utilizado en el primer análisis para los tres corpus. Por lo que se ha hecho un análisis de 68 vídeos.

Este análisis se realiza para comprobar si realmente los nombres que está esperando el fichero de referencia, la herramienta de detección de texto *(LOOV)* detecta estos nombres.

Los resultados que se han obtenido de este análisis son los siguientes:

En primer lugar, si nos centramos en un vídeo x, en el fichero de referencia aparece un nombre y_unas cuántas veces, por lo que en el fichero OCR de ese vídeo debería de aparecer ese nombre aproximadamente las veces que sale en el fichero de referencia, pero sin embargo este nombre y aparece muchas menos veces o solo una vez.

En segundo lugar, se ha observado que en el fichero de referencia existen nombres que en los ficheros OCR no detectan, sobre todo para el corpus de INA.

En tercer lugar, los ficheros OCR detectan más nombres de los que existen en el fichero de referencia. Esto afecta a que cuando se realiza la extracción de nombres, se va a detectar nombres que en el fichero de referencia no existen para ese vídeo.

En cuarto y último lugar, se ha observado que en el fichero de referencia no se encuentran todos los vídeos que hemos analizado.

Presupuesto

5

5. Presupuesto

Horas dedicadas al desarrollo del proyecto: 600

Work Packages	Horas dedicadas
• Propuesta del proyecto y plan de trabajo	30
• Lectura del estado del arte	50
• Investigación algoritmos	60
• Desarrollo Stanford NER	150
• Desarrollo de un algoritmo a través de filtraje	120
• Evaluación de los resultados y propuesta de mejoras	100
• Redacción de la memoria final	90

Para calcular el presupuesto también se debe de tener en cuenta que el trabajo se ha realizado en un servidor del grupo de imagen y vídeo de la Universidad Politécnica de Catalunya, el cual tiene las siguientes características:

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/Unix Usage
g2.2xlarge	8	26	15	60 SSD	0.70€/h

Tabla 8 Detalles del servidor Imagen/Vídeo de la UPC

Se ha utilizado el servidor desde que se comenzó a desarrollar el proyecto, por lo que se han dedicado 450h aproximadamente.

Además se ha dedicado 1h cada semana aproximadamente para la supervisión del trabajo por parte del tutor, siendo 30 semanas.

Con todos estos datos, y sabiendo que un ingeniero junior debería de cobrar 10€/h de media, la supervisión del trabajo por parte de un ingeniero senior se paga a 40€/h, se puede concluir que el valor del presupuesto sería de 7515€ aproximadamente.

Conclusión

6

6. Conclusión

Los objetivos que se habían marcado para la creación de este proyecto era el desarrollo de nuevos algoritmos que ayudaran a la mejora de extracciones de nombres respecto al desarrollo de años anteriores.

Los resultados que se han obtenidos a través de la métrica *MAP* no son comparables con años anteriores, ya que el fichero que teníamos como referencia no contenía todos los vídeos y le faltaban muchos nombres en cada vídeo de los cuales en nuestros resultados sí que obteníamos.

Después de hacer el análisis más profundo y la obtención de resultados, se puede comprobar que los dos algoritmos que se han desarrollado en este proyecto sí que realizan bien su función y por lo tanto filtran bastante bien.

Se puede confirmar que el algoritmo de Stanford NER funciona mejor que el algoritmo basado en diccionarios de nombres y apellidos.

De cara a la mejora del desarrollo de este proyecto, se puede plantear las siguientes ideas:

- Mejorar las técnicas de detección de texto y probar el desarrollo que se ha mencionado en el apartado del estado del arte (*CTPN* del repositorio de github *Deep Video Analytics*).

- Volver a revisar los diccionarios que pertenecen a los nombres, apellidos, palabras negativas y palabras neutrales. Añadiendo nombres y apellidos de lenguas extranjeras, sobre todos para el idioma francés.

Bibliografía

[1] Benchmarking multimedia technologies with the CAMOMILE platform: the case of Multimodal Person Discovery at MediaEval 2015. (2016). En J. Poignant, H. Bredin, C. Barras, M. Stefas, P. Bruneau, & T. Tamisier, *In Proceedings of the 10th LREC Language Resources and Evaluation Conference.* Portoroz, Slovenia.

[2] Everingham, M., Sivic, J., & Zisserman, A. (s.f.). "Hello! My name is... Buffy". *Automatic Naming of Characters in TV Video*.

[3] Face recognition from caption-based supervision. (s.f.). En M. Guillaumin, T. Mensink, J. Verbeek, & C. Schmid, *Institut National de Recherche en Informatique et en automatique.*

[4] Freeling. (s.f.). Obtenido de https://talp-upc.gitbooks.io/freeling-4-0-user-manual/content/

[5] Freeling. (s.f.). Obtenido de http://nlp.lsi.upc.edu/freeling/node/1

[6] Freeling. (s.f.). Obtenido de https://legacy.gitbook.com/book/talp-upc/freeling-4-0-user-manual/details

[7] Fusion of speech, faces and text for person identification in TV broadcast. (s.f.).

[8] MediaEval. (s.f.). Obtenido de http://www.multimediaeval.org/

[9] Multimodal Person Discovery in Broadcast TV at MediaEval 2015. (2015). En J. Poignant, H. Bredin, & C. Barras, *Working Notes Proceedings of the MediaEval 2015 Workshop.* Wurzen, Germany.

[10] Multimodal Person Discovery in Broadcast TV: lessons learned from MediaEval 2015. (2016). En J. Poignant, H. Bredin, & C. Barras, *Submitted to IEEE Transactions on Multimedia.*

[11] Multimodal Understanding for Person Recognition in Video Broadcasts. (2014). En F. Bechet, M. Bendris, D. Charlet, G. Damnati, B. Favre, M. Rouvier, . . . G. Senay, *Interspeech 2014, Fifteenth Annual Conference of the International Speech Communication Association.*

[12] Person Instance Graphs for Named Speaker Identification in TV Broadcast. (2014). En H. Bredin, A. Laurent, A. Sarkar, V.-B. Le, S. Rosset, & C. Barras, *Odyssey 2014: The Speaker and Language Recognition Workshop.*

[13] Poignant, J., Besacier, L., Quénot, G., & Thholland, F. (2012). From text detection in videos to person identification. *IEEE International Conference on Multimedia and Expo*.

[14] Proyecto Camomile. (s.f.). Obtenido de https://camomile.limsi.fr/doku.php

[15] Proyecto Camomile. (s.f.). Obtenido de http://www.chistera.eu/projects/camomile

[16] QCompere REPERE 2013. (2013).

[17] Repère. (s.f.). Obtenido de http://www.defi-repere.fr/

[18] Stanford NER. (s.f.). Obtenido de https://nlp.stanford.edu/software/CRF-NER.html

[19] Stanford NER. (s.f.). Obtenido de https://stanfordnlp.github.io/CoreNLP/

[20] The first official REPERE evaluation. (2013). *Proceedings of the First Workshop on Speech, Language and Audio in Multimedia (SLAM)*.

[21] The First Official REPERE Evaluation. (2013). En G. Bernard, O. Galibert, & J. Kahn, *SLAM 2013, First Workshop on Speech, Language and Audio for Multimedia.*

[22] The REPERE Corpus: a Multimodal Corpus for Person Recognition. (2014). En A. Giraudel, M. Carré, V. Mapelli, J. Kahn, O. Galibert, & L. Quintard, *LREC 2014, Eighth International Conference on Language Resources and Evaluation.*

[23] Unsupervised Speaker Identification in TV Broadcast Based on Written Names. (2015). En J. Poignant, L. Besacier, & G. Quénot, *IEEE/ACM Transactions on Audio, Speech, and Language Processing.*

[24] UPC. (2016). *Repositiorio Github.* Obtenido de https://github.com/imatge-upc/mediaeval2016

[25] UPC. (s.f.). *Image Processing Group*. Obtenido de https://imatge.upc.edu/web/publications/towards-large-scale-multimedia-indexing-case-study-person-discovery-broadcast-news

[26] CTPN. (s.f.). Obtenido de http://slade-ruan.me/2017/10/22/text-detection-ctpn/

[27] CTPN. (s.f.). Obtenido de https://arxiv.org/abs/1609.03605

[28] MAP. (s.f.). Obtenido de https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html

[29] LOOV. *Repositiorio Github.* Obtenido de https://github.com/johannpoignant/LOOV

[30] Deep Video Analytics. *Repositiorio Github*. Obtenido de https://github.com/AKSHAYUBHAT/DeepVideoAnalytics

Listado de ilustraciones

Listado de Tablas